Introduction
Mark Redekopp
What is Computer Science

• All science is computer science
  – It is very interdisciplinary: Math, Engineering, Medicine, Natural sciences, Art, Linguistics, etc.

• Computer science is no more about computers than astronomy is about telescopes

• It is about solving information-based problems using computers as the main tool
  – How do I recognize objects in a photograph
  – How do I determine the best web page to return given a search query
  – Identify the function of this protein given it structure
What Computer Scientist Do…

• Observe, organize, transform and discover useful information from data
• Use math and logic to solve problems
• Work in groups
• Innovate and improve
• Program
Computer Science Is…

• Essential to economic growth of our country
• Dealing with society’s problems
  – Bioinformatics and E-Science
  – Big Data
    • Conservation & the environment
    • Developing personalized learning
• A great way to make a living
  • http://www.huffingtonpost.com/maria-klawe/computing-our-childrens-f_b_388874.html
What Is this Course About

• Introduction to Programming
  – Introduction: Don't expect prior programming experience
    • However, we will move fast so you must be prepared to put in some extra time if you've never coded before
  – Programming
    • We'll focus on the C/C++ language (not Java)
    • We'll try to teach good coding practices and how to find efficient solutions (not just any solution)
High Level Languages

Just like half of the world's spoken tongues, most of the 2,500 plus computer programming languages are either endangered or extinct. As powerhouses C/C++, Visual Basic, Cobol, Java and other modern source codes dominate our systems, hundreds of older languages are running out of life.

As a collection of engineers-electronic linguists, if you will, we aim to save, at least document the lingos of classic software. They're combing the globe's 9 million developers in search of coders still fluent in these nearly forgotten frangues. Among the most endangered are Ada, APL, B (the predecessor of C), Lisp, Oberon, Smalltalk, and Simula.

Code-caker Grady Booch, Rational Software's chief scientist, is working with the Computer History Museum in Silicon Valley to record and, in some cases, maintain languages by writing new compilers so our ever-changing hardware can speak the code. Why bother? "They tell us about the state of software practice, the minds of their inventors, and the technical, social, and economic forces that shaped history at the time," Booch explains. "They'll provide the raw material for software archaeologists, historians, and developers to learn what worked, what was brilliant, and what was an utter failure." Here's a peek at the strongest branches of programming's family tree. For a nearly exhaustive rundown, check out the Language List at HTTP://www.informatik.uni-freiburg.de/Java/misc/language_list.html. - Michael Mendeece

Key
- Year Introduced
- Active: Thousands of users
- Prototyped: Brigham University; compilers available
- Prototyped: Large dropping off
- Extinct: no known active users or obsolete compilers

Survival of the Fittest
Reasons a language endures, with examples of some classic tongues
Appeals to a wide audience C (driven by the popularity of Unix)
Gets a job done Cobol (designed for business-report writing)
Delivers new functionality Java (runs on any hardware platform)
Fills a niche Mathematica (speeds up complex computations)
Offeras a mojic of elegance Icon (has friendly, line-oriented syntax)
Has a powerful base or backing C# (developed by Microsoft for .Net)
Has a charismatic leader Perl (programmer-author Larry Wall)
Why C++

• C/C++ is close to the actual hardware
  – Makes it fast
  – Makes it flexible (Near direct control of the HW)
  – Makes it dangerous (Near direct control of the HW)
  – C became popular because it was the language used to implement Unix & then Linux (all Unix/Linux distributions came with a C / C++ compiler)

• C/C++ is ubiquitous
  – Used everywhere, even to implement other programming languages (i.e. Python, Matlab, etc.)

• Wide availability of compilers and development environments
Syllabus
Course Advice

• Catch the wave!
  – Underestimate the time you will need and your ability to get your work done
  – Limit extracurricular activities in the 1st semester
  – Don’t let shame or embarrassment keep you from the help you need

• You’re here to learn not to be taught
  – Be active and engaged
  – Do not be afraid to fail
  – [http://ceng.usc.edu/~bkrishna/TheDangersOfClassroomTeaching.pdf](http://ceng.usc.edu/~bkrishna/TheDangersOfClassroomTeaching.pdf)

• Computer science requires practice
  – It's like learning a musical instrument
Four men are buried up to their necks by a firing squad and are told they have two minutes for someone to yell out the color of their own hat. If they say nothing, they will all be shot. The men know that there are two black hats and two white hats. Man 1 and man 2 can only see the brick wall. Man 3 can see Man 2 and the brick wall. Man 4 can see Man 2, Man 3, and the brick wall. They are not allowed to talk to each other. Which man will know the color of his hat and be certain he is right?
Solution: If Man 4 sees 2 white hats or 2 black hats in front of him he knows right away the color of his hat. If he sees different colors in front of him, he cannot deduce what color he is. However, if that is the case, then man 3 upon not hearing an answer from man 4 should realize man 4 doesn’t have enough information to know the color which means man 2 and 3 have different color hats. Thus man 3’s hat must be the opposite of the color he sees on man 2. He can shout out his hat color with certainty.
What’s the Difference in a Major?

**CS**
- Degree offered by the college with slightly more freedom in electives
- **Goal**: To train engineers who can develop and design today’s complex software & information systems and innovate tomorrow’s computing applications
- Variations:
  - CSBA, CSGM

**CECS**
- A joint degree between the CS and EE departments of USC
- **Goal**: To develop engineers who can span the complex inter-relationship of computer hardware and software, creating and designing system solutions
CECS Curriculum

Core Concepts (Programming Languages, Data Structures, Digital Logic Design)

- AI (CS 445, CS 460)
- Operating Systems (EE 101, EE 201, EE 109)
- Software Engineering (CS 104, CS 201, CS 210)
- Graphics (CS 377, CS 477)
- Computational Theory / Algorithms (CS 377, CS 477)
- Computer Architecture (EE 457, EE 454)
- Networks & Web Programming (EE 101, EE 328)
- EE 450 (EE/CS 450, CS 351, EE 477)
- EE 479 (EE/CS 450, CS 351, EE 477)
- EE 459 (EE/CS 450, CS 351, EE 477)
- EE 454 (EE/CS 450, CS 351, EE 477)
- EE 457 (EE/CS 450, CS 351, EE 477)
- EE 457 (EE/CS 450, CS 351, EE 477)
Research at USC

- Integrated Media Systems Center
  - Sound, video, online collaboration, streaming media research
- Information Sciences Institute
  - AI, Internet, Advanced Processing Systems research
- Institute for Creative Technologies
  - Virtual Reality, Graphics, Animation, Games
Media

• Robotics
  – http://www.isi.edu/robots/superbot/movies/FoxNews.swf

• Virtual Reality
  – http://www.youtube.com/uscict#p/u/13/Fh9glSwxbvU
  – http://www.youtube.com/uscict#p/u/0/0U7-q_9YV5c
THINK LIKE A COMPUTER
Path Planning

• Find shortest path from S to F
Path Planning

• Find shortest path from S to F
Path Planning

- A computer usually can only process (or see) one data item (a square) at a time

May just compute a straight line path from ‘S’ to ‘F’
Path Planning

• What if I don’t know where the Finish square is? Now I want to find ‘F’ and the shortest path to it while examining the minimum number of squares as possible.
Path Planning

- Examine all closer squares one at a time before progressing to further squares.

If you don’t know where F is and want to find the shortest path, you have to do it this way.

Uninformed search for shortest path: *Breadth-first*
Path Planning

- Now I’ll tell you where F is
- Can that help you reduce the number of squares explored?

Select a square to explore with minimum distance to the finish.
Path Planning

• Now I’ll tell you where F is
• Can that help you reduce the number of squares explored?

Select a square to explore with minimum distance to the finish.
Path Planning

• But what if we run into a blockage?
  – Now we would pick the best among the remainder.

Select a square to explore with minimum distance to the finish
Path Planning

- But what if we run into a blockage?
  - Now we would pick the best among the remainder.

Select a square to explore with minimum distance to the finish
• Why can’t computer just “look” at the image
  – Computer store information as numbers
  – These numbers are stored as units of 8-, 32- or 64-bits and the processor is only capable to looking at 1 or 2 numbers simultaneously
  – Each pixel of the image is a separate piece of data

Processor

RAM

32-64 bits
Memory

- Set of cells that each store a group of bits (usually, 1 byte = 8 bits)
- Unique address assigned to each cell
  - Used to reference the value in that location

<table>
<thead>
<tr>
<th>Address</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>11010010</td>
</tr>
<tr>
<td>1</td>
<td>01001011</td>
</tr>
<tr>
<td>2</td>
<td>10010000</td>
</tr>
<tr>
<td>3</td>
<td>11110100</td>
</tr>
<tr>
<td>4</td>
<td>01101000</td>
</tr>
<tr>
<td>5</td>
<td>11010001</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>1023</td>
<td>00001011</td>
</tr>
</tbody>
</table>

Memory Device
Memory Operations

- Memories perform 2 operations
  - Read: retrieves data value in a particular location (specified using the address)
  - Write: changes data in a location to a new value

- To perform these operations a set of **address**, **data**, and **control** inputs/outputs are used
  - Note: A group of wires/signals is referred to as a ‘bus’
  - Thus, we say that memories have an **address**, **data**, and **control** bus.
Programming vs. Algorithms

- Programming entails converting an algorithm into a specific process that a computer can execute.
20-Second Timeout

• Who Am I?
  – Teaching faculty in CENG
  – Undergrad at USC in CECS
  – Grad at USC in EE
  – Work(ed) at Raytheon
  – Sports enthusiast!
    • Basketball
    • Baseball
    • Ultimate Frisbee?
Your Environment

GETTING STARTED
Development Environment

• To write and run software programs in C you will need
  – A text editor to write the code
  – A ‘C/C++’ compiler, linker, etc. to convert the code to a program

• Different OS and platform combinations have different compilers and produce “different version” of a program that can only run on that given OS/platform.
  – Mac XCode (Mac only)
  – MS Visual Studio (Windows only)
  – CodeBlocks (cross-platform)
Ubuntu VM Image

• We are providing a virtual machine appliance (An Ubuntu Linux image that you can run on your Mac or Windows PC)
  – Requires installation of Oracle VirtualBox and download of the Ubuntu Image
  – Requires download of the VM Image
    [http://ee-classes.usc.edu/cs101/course-vm.ova](http://ee-classes.usc.edu/cs101/course-vm.ova)

• Video walkthrough